The Lingua Parallel Concordancing Project: Managing Multilingual Texts for Educational Purpose

Laurent Romary, Nathalie Mehl, David Woolls

The Lingua Parallel Concordancing Project: Managing Multilingual Texts for Educational Purpose

Laurent Romary*, Nathalie Mehl* and David Woolls†

*CRIN-CNRS & INRIA Lorraine
Bâtiment Loria, B.P. 239
F-54506 Vandœuvre Lès Nancy
e-mail: romary@loria.fr

†University of Birmingham
Birmingham B15 2TT
e-mail: DAVID@ccbl.bham.ac.uk

Summary

This paper describes some technical aspects of the Lingua Parallel Concordancing Project which aims at managing a multilingual corpus to ease students' and teachers' work in second language learning. More specifically we show some implementation issues of the Text Encoding Initiative guidelines and the corresponding tools which we have developed.

Introduction

A multilingual corpus for parallel concordancing

The proposal for a parallel concordancing project put to the Lingua bureau of the EU originated in the desire of a group of lecturers from different European universities[1] to enhance the use of concordancing in the process of learning a second language. Concordancers have already been widely used in teaching (Tribble & Jones, 1990; Johns & King, 1991). The concordancer can be used by a teacher to obtain teaching material relating to specific grammatical content or used by a student for free exploration of the use of a word in context, and in many ways in between. Concordancers are built into the large corpora of English language held by Cobuild and the British National Corpus. They are also available for the smaller user in programs such as the do-it-yourself approach of Tribble & Jones and in commercial programs such as MicroConcord (Johns 1986). University users have long had access to the Oxford Concordance Program. What all these have in common is that they are designed to operate in a momolingual environment, even though they may be capable of handling more than one language.

Multilingual corpora are also currently being built for exploration of language. Notably, another EU project, Multext, is building such a corpus with the express intention of making it available for the development and assessment of tools for multilingual text handling. This will contain some parallel texts, but only as a small component of a wide range of texts in many languages.

---

[1] The University of Nancy II (France), the University of Birmingham (UK), the University of Hull (UK), the University of East Anglia (UK), the University of Wuppertal (Germany), the University of Turin (C.L.A.U., Italy), the University of Patras (Greece), the University of Arrhus (Danemark).

However, to perform parallel concordancing one clearly requires parallel texts. The pedagogic aims of the project on which we are working also have this requirement, with the added constraint that all the translations used must be of sufficient quality to provide accurate information for teachers and learners. The identification, assessment and collection of these texts is outside the scope of this article, but an indication of the range of material is given to indicate why such a corpus is desirable. The texts in progress include children's literature, general fiction, a play, journalism and technical manuals. The six languages of the initial project are Danish, English, French, German, Greek and Italian.

The purpose of such a diversity of literature is to provide as wide a range of examples of translation of words and phrases in different environments as is possible within the constraints of a corpus of what will always be of a much smaller size than the monolingual corpora size. It is also to allow exploration to be restricted to one or other grouping or genre. It is hoped that the corpus will provide material for teachers at all levels of education.

For those of us creating the management system for such a corpus, the variety of text types and conventions provides a major administrative problem, and it is because of this that it was decided to recommend the application of the TEI guidelines to the project. We need to be able to identify and differentiate between source languages and translations, group into genres, acknowledge copyrights and above all align the texts, and provisions for all of these elements and more are included within the TEI guidelines. To have attempted to design our own system would have been adventurous, to say the least. It seems to the authors, if not necessarily to all our colleagues, that the considerable effort in applying the guidelines to the corpus, will allow us to produce a working concordancer much sooner. For those intending to follow this route, it should be pointed out that presenting a fully encoded text to a group of linguists can be a nerve-wracking experience as you try to explain why you have trebled the length and made it totally unreadable in the process.

Making the corpus manageable

As indicated above, in a multilingual corpus there is no real alternative to modifying the plain text as presented by the various sites. We will discuss below four of the major components in the corpus maintenance process; header management, character encoding, paragraph and sentence slicing, and text alignment. The first allows the classification and identification of documentation to meet international standards and makes the corpus accessible to others in the long term. It also will allow the use of material produced by other research centres and institutions who have been working on TEI. The second allows text to become machine readable irrespective of the source of the input or the desired method of output, by applying the rules of the TEI/SGML syntax to the text. The third is necessary both for the alignment process to oprate and to allow flexible and clear concordance presentation. The fourth is the structure on which the retrieval process can operate. By combining the information of these four elements words and phrases can be retrieved with any amount of context, for particular groupings of texts, in any selected pair of languages. This amount of information and degree of flexibility cannot be obtained from a monolingual

concordancer operating on plain text. The different applications require very different approaches to corpus management.

Different steps in the archiving process

Header Management

A header is supposed to allow a speedy introduction to the remaining document. At a glance one has to obtain a general idea of what the document is about and how it is rendered.

The TEI working committee has recommended putting a header in all textual documents. The TEI guidelines give a very detailed explanation about how to use, to build and optimize the header. Apart from a few elements which cannot be missed out according to their obligatory status, the header construction seems very permissive and may take any shape depending on who it is made for and what kind of information it should bear. To be more precise, the TEI offers either a nearly infinite list of tags with unique meaning which produces a nearly infinite degree of depth of the header or alternative tags such as **<p>** allowed to contain information of any kind for which particular tags are not provided explicitly by the latest publication of the guidelines. Such permissive rules of construction allow remarkable extensions to the potential use of the header.

The TEI compulsory **<teiHeader>** element mainly gathers bibliographic information. It supplies the user with quick general information on the contents of the electronic file such as the **<title>**, the **<author>** and its **<extent>** (i.e. size). These are the user's first criteria of choice. If the user doesn't know the works or novels available in the corpus, his choice may be influenced by the list of **<keywords>** found in the header and giving a text or text class description. For example let us imagine that a non-English-speaking scientist, researcher, engineer or whatever his professional position is, is working on an English report of a international conference on polymers and needs the translation equivalent to the verb 'to dilute'. He might have more chance of finding the proper meaning in already translated chemical reports, books or tutorials rather than in literary documents such as Shakespeare plays.

The multilingual aspect of a text is not taken explicitly into account in the current header. The **<langUsage>**, **<language>** and **<lang>** elements as well as the *lang* attribute rather refer to foreign languages, for exemple latin quotations, existing within the current document.

In the context of the Lingua project, before choosing a text the user first has to check if it exists in the language he is interested in. An element describing the different existing translations submitted to the alignment program mentioned in the TEI guidelines in chapter 14.4.2. Alignment of parallel texts would be very useful for our purpose. The following proposals of additional tags have already been made to the TEI responsibles :

<translations>

<translation>

```
            <language>EN</language>
            <translator>J.Smith</translator>
        </translation>
        <translation>
            <language>FR</language>
            <translator>M.Dupond</translator>
        </translation>
</translations>
```

We leave it to the working committee to judge whether such elements are sufficient for the TEI users or not. For the moment in order to stay TEI conformant we decided to include this information in a **<respStmt>** element as follows, while waiting for more explicit elements to be created :

```
<respStmt>
        <resp>translated by
            <lang>EN</lang>
        </resp>
        <name>J.Smith</name>
</respStmt>
```

The TEI header also gives information which is additionally useful for our project. It tells us where the electronic file comes from and who worked on it before it was imported on our local network. It says what kind of encoding has been used and what kind of changes the electronic document had to undergo before release to other site.

The first stage of our parallel multilingual concordancing sofware consists in word or phrase searching. The user enters a word or phrase which he wants the translation of in a input field of the Mosaic[2] interface and the software browses through a corpus of texts and looks for the wanted expression. This does not pose any problem as long as the corpus of texts is not too large. But this kind of problem comes up very soon. The user will see himself obliged to choose the texts where he wants the translations to be taken from. Our role is to give the user an easy way of making the right choice of texts. Thus, if the user of the file needs further explanations about encoding methods, the reason of omitted passages or anything concerning the document layout, he has the opportunity to ask the appropriate questions and discuss his problems with people who dealt with the file before him.

---

[2] *NCSA Mosaic* is an Internet information browser and World Wide Web client, which is available on different computer plateform. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign. This program can be used as a simple interface builder in the case where small modules are being develop as it is the case in corpus management.

What we intend to do first is associate a header to each text, always keeping in mind that common parts of text headers may be gathered in a corpus header in the long run. We use a home-made program running under Mosaic to create an interface allowing us to display our header and change anything we want in it. This tool seems to work fine but is still being tested and improved. Thus, if we decide to change the way of slicing our text, we will inform any other user about it by changing the content of the element reporting about this kind of changes, that is **<encodingDesc>**.

Character Encoding

Dealing with character encoding in the context of a multisite, multilingual corpus leads to several problems which are difficult to tackle within one single framework. The problem has been considered in the TEI guidelines since one of the goals of TEI is to facilitate extensive interchange of textual data.

Leaving aside the problem of keyboard encoding, which is usually kept transparent for most users, we can distinguish three main steps in the life of a character within a given piece of textual data[3]: it is first scanned or typed in at a given site, added to the general corpus and finally visualized for the specific purposes of the project.

SGML provides a general way of encoding characters in a text on the basis of a 7-bit ascii code (ISO 646). The corresponding character set includes all graphical characters and special characters together with the tabulation (\t) and carriage return (\n) characters. Any character belonging to this set will appear unchanged in the corpus. On the contrary, language specific characters have to be encoded specifically by means of a SGML entity started by an '&' and ended by an ';'. For example, 'é' will be encoded as `é`. Two characters are reserved for marking purpose, '<' and '&' which have to appear respectively as `&inf;` and `&` if not used for encoding purposes. This means that the different sites where the text of our corpus are acquired should not think of doing home made SGML marking since the corresponding tags would be encoded during the integration process. Since each site may be using a specific character set, it has been necessary for us to collect or build up the corresponding transfer tables to SGML entities[4]. These tables can now handle texts coming from PCs (tables 437, 850, 851, elot 928 for Greek), Macintoshs and other platforms using standards such as ISO8859-1 or ISO8859-7 (again for Greek…).

---

[3] At least, the languages we have to deal with in our project are restricted to alphabetical ones. This means that the corresponding alphabets are each based on a limited number of characters which can be encoded by means of a one byte representation, as opposed to languages such as Japanese which requires at least two.

[4] the TEI now provides Writing System Declarations for most of the usual encoding tables.

Here below is an example of character encoded text taken from St Exupéry's *Le petit prince* in French and in Danish.

Original texts

J'ai donc dû choisir un autre métier et j'ai appris à piloter des avions. J'ai volé un peu partout dans le monde. Et la géographie, c'est exact, m'a beaucoup servi. Je savais reconnaître, du premier coup d'oeil, la Chine de l'Arizona. C'est très utile, si l'on est égaré pendant la nuit.	Jeg blev nødt til at vælge en anden bestilling, og jeg lærte så at flyve flyvemaskiner. Geografien har ganske rigtigt været mig til stor hjælp. Jeg kunne med et eneste blik kende forskel på Kina og Arizona. Og det er meget praktisk, hvis man er fløjet vild om natten.

Encoded version

J'ai donc dû choisir un autre métier et j'ai appris à piloter des avions. J'ai volé un peu partout dans le monde. Et la géographie, c'est exact, m'a beaucoup servi. Je savais reconnaître, du premier coup d'oeil, la Chine de l'Arizona. C'est très utile, si l'on est égaré pendant la nuit.	Jeg blev nødt til at vælge en anden bestilling, og jeg lærte så at flyve flyvemaskiner. Geografien har ganske rigtigt været mig til stor hjælp. Jeg kunne med et eneste blik kende forskel på Kina og Arizona. Og det er meget praktisk, hvis man er fløjet vild om natten.

The second important aspect in character management is the visualization of a text from its encoded representation. Whereas adopting a uniform scheme for the centralization of a corpus is quite a simple task, it is clear that we are currently far from achieving a general interface which will be able to show any kind of language on any kind of computer platform. However, it is possible to put forward the main issues related to this problem. The first thing to do for any part of a given text to be viewed is to determine the language it is made of. This can be computed through the inheritable *lang* attribute which has to be kept in memory during any search within a SGML tree. By the way, the TEI guidelines has nicely declared this attribute as a global one which may thus appear together with any SGML tag. The second step should then decode the different entities appearing in the textual segment under consideration by using the proper reverse table depending on the the target computer platform on which it is to be viewed. Finally, one has to find a complete set of fonts for the different platform which may be used for viewing.

Paragraph and Sentence Slicing

There are two main reasons why sentence and paragraph identification is necessary in a multilingual corpus. The first relates to alignment and the second to the user interface.

Alignment needs boundaries within which to work. Depending on the genre of the texts broad divisions are available; chapters in fiction, sections in documentation, sub-headings in newspapers and acts and scenes in plays. Guidelines exist for the markup of these features, but the units are normally still too large for the degree of alignment required for pedagogical purposes. Narrower divisions into paragraphs are common to most texts. However, unlike the grammarians who need to mark every word, the depth of markup required by concordances can stop at sentence level, since such tools are designed to encourage observation and exploration of particular words in a variety of reasonably complete contexts.

The objective of the project is to automate as much of the markup process as is practicable. The majority of texts are being scanned in, and by setting the appropriate parameters, it is possible to produce input text which contains end-of-line indicators only at paragraph boundaries. Word-processed files have this format by default. It is therefore generally possible to replicate the paragraph boundaries for such texts quite easily.

Sentence boundaries offer much more of a challenge. Here we are not simply concerned with '. ? !' as terminators, but their combination with) " ' or with each other as in ellipsis ... or !!! With the addition of dialogue and differing orthographic conventions, such as the Greek question mark ; or the European quotation marks « » , the identification process needs to be both sensitive and robust. SGML conventions are designed to assist in such difficulties by describing the quotation marks, for instance, rather than reproducing them. So « becomes « standing for left angled quotes as opposed to " which is " and ` which is ‘ left single quote. By applying the SGML codes before attempting to split sentences, some of the problems are removed. SGML codes also exist for marking non-terminating full stops, such as occur in percentages '2.4%' or biblical chapter references John 1.4. The last sentence ends with a pair of numbers both followed by full stops but only one of which ends the sentence. This makes for entertaining programming.

Inevitably some texts impose far more strain on an automatic process than others. Not all texts finish a paragraph with one of the normal sentence terminators. Both the French and English copies of Le Petit Prince have paragraph separation after a colon followed by another paragraph containing the direct speech in inverted commas. Because of the need to preserve the layout, this means that for this text colons followed by a line break are defined as sentence terminators. We have had to set limits on what is and is not acceptable to avoid having mutually exclusive conditions or extensive post-editing. For alignment purposes, as long as the routines are applied consistently across the languages, the shorter sentences sometimes produced are an advantage. This is not so true of the user interface.

We have used the standard SGML codes for paragraphs <p> and sentences <s> together with attributes for unique identifiers for both. An example paragraph is shown below.

<s id=p4s1 n=24>This is the first sentence.</s><s id=p4s2 n=73>The second sentence contains a quotation which looks like this &quote;What a strange language&quote;.</s><s id=p4s3 n=36>That's quite enough of this illustration.</s>

The additional attribute of n=24 in the first sentence start tag is the number of orthographic alpha-numeric characters in the sentence. This is included to allow the alignment program to read only the sentence element information to decide whether the sentences are in direct or combined alignment, or are missed out altogether. The items starting with & and finishing with ; are descriptions of the start and end quotes and the apostrophe respectively. This allows the transcription of quotes in Greek to <<>> and in English to " " simply by appropriate translation tables. As this is a strict SGML implementation , all paragraphs and sentences are terminated with and </s> respectively.

Text alignment

One of the central point in the process of our project is the alignment of a given text with its different translations. There has been very few works in this domain and one usually refers himself to Gale and Church's (Gale & Church, 91) work where we can find an original application to this problem of an algorithm of synchronization sometime known as Dynamic Time Warping (Fu, 82; Miclet, 86). As a matter of fact, aligning a text and its translation amounts to finding the better linear match between chunks of texts corresponding to a given level of description, that is sections, paragraphs or sentences. The alignment process is schematized in figure 2 where it may be observed that there is no necessity for a one to one match between sentences for example, but at times two sentences may be translated into one or on the contrary a given sentence can be split into two.

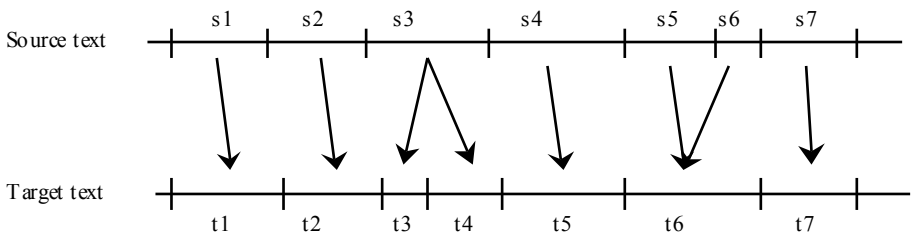

Figure 2

It is not our purpose here to describe the algorithm as such but much more to show how the results obtained from it may be used in the TEI framework. In chapter 14 *Linking, Segmentation, and Alignment*, TEI P3 proposes several different ways to encode such alignments as those produced by our algorithm. Indeed, considering the different constraints we had chosen to maintain on our corpus, one coherent solution appeared to us. First, we wanted to distinguish in our encoding scheme what would belong to alignment marking from the text itself. We thus rejected to introduce any specific marking such as <seg>

or <anchor> tags[5]. Besides, as the different translations are kept within differents files (and thus different SGML documents), it appeared to us that the useful device provided by external pointers would be perfect for our purpose. Finally, to ease the retrieval of any textual part in our corpus we made the assumption that any segmental element would bear an identifier attribute/value pair (<*id*> tag), thus providing an easy way to build textual segments by means of the <*link*> tag.

From the different constraints mentioned above, it seemed natural to come to the encoding scheme which is exemplified by the example given in the appendix. This scheme has the following characteristics:

• any information concerning text alignment is exclusively marked within the source text, thus ensuring that it keeps a central point for any further concordancing operation;

• the alignment information appears within the span of the <*text*> tag but outside the *body* tag.

Concerning the SGML marks as such, we can distinguish three parts in the encoding scheme. The first one corresponds to external pointers to the target text. For instance, <*xptr id=xptr1 doc=target.TEI from='ID(p10s1)'*>creates a new *id* within the source text which corresponds to sentence *p10s1* in the target text. The second section of the alignment tags contains possible concatenations of textual elements, when more than one sentence has been aligned by our algorithm for example. These concatanetations may either involve tags from the source text or from the target text since the latter has been made accessible through the external pointer mechanism. Finally, a <*linkgroup*> puts together the differents pairs of source/target segment associations (e.g. <*link targets='p10s1 xptr1'*>).

Parallel Concordancing

As mentioned above, for our concordancer we are primarily interested in the behaviour of words in context across different languages. Unlike KWIC (Key Word in Context) concordancers, which show the search word centralised in a single line of text, the preferred format for a multilingual display is the sentence. This is mainly because although the context word is known in the search language, there is no way of knowing where in the target language paragraph the relevant translation word will appear, if indeed it appears at all. As described in the alignment section of this paper, there is the possibility that the required word or words will appear in a preceding or following sentence, rather than the equivalent single sentence of the search language. The identification of shorter sentences by the automatic method may lead to the selection of samples which are artificially brief and therefore insufficiently informative. Partly to cover this problem an intermediate stage of paragraph display will be available as well as the full text offerred by KWIC concordancers. This is facilitated by the SGML elements and attributes for paragraph and sentence identification.

---

[5] The <seg> tag may be used to put together a given *segment* of a document so that it might be linked with another segment, for example in the target text. The <anchor> tag marks a direct pointing towards another SGML node, with the drawback that the span of the actual alighnment is not explicit.

The identifiers are used in creating a particular concordance. Unlike monolingual concordancers, it is not sufficient simply to find the search word and display it in its immediate context. It is also necessary to find out which in which sentence each instance occurs, look for a cross reference in the alignment table and then find the relevant sentences in the target language. The two languages can then be displayed side by side. In fact, it is proposed to use a KWIC concordance display for the search language. This has the advantage of allowing the user to see the maximum number of examples on screen and sorting them to left or right using monolingual techniques, and to be able to move to the side-by-side presentation from any given line.

Regrettably, the existence of the codes in the text which allows this to happen disrupts the natural method of building concordance lines, as the formatting gets in the way of simple pointer arithmetic within the text file. A linear concordancer has been built which deals with this difficulty, but it is expected that a fully inverted file method will be designed which would simplify line, sentence and paragraph reconstruction considerably. The location ladder built into the TEI document in chapter 14 is being explored as a potential basis for this approach. As the concordancer program is written in C++ the various elements and their attributes can be placed in different objects defining a line, a sentence and a paragraph for simple re-construction or manipulation.

Data retrieval and User interface

A toolbox oriented management of the corpus

As shown in figure 3, a given text goes through a whole process of transformations before appearing in a usable form within our corpus. Indeed we can distinguish two main phases in this process, one corresponding to a preprocessing working on raw data and yielding a SGML compatible document and one progressively adding new information to text by simply adding new SGML elements to it or making up existing one. As shown in this paper this second phase makes it possible to consider a whole set of independent modules specialized for a specific type of information and the corresponding treatments. Even if they can be independently conceived, all those modules share at least two main characteristics. First, they have been built upon a set of SGML handling functions available within the Dilib library already developed within the Centre de Recherche en Informatique de Nancy[6]. Second, as already mentioned, they are all accessible within the common interface provided by *Mosaic* on the Unix/XWindow platform which is currently used for the corpus management.

[6] This library, first intended to cope with bibliographical data has been developped under the respondsability of Jacques Ducloy (Jacques.Ducloy@loria.fr).

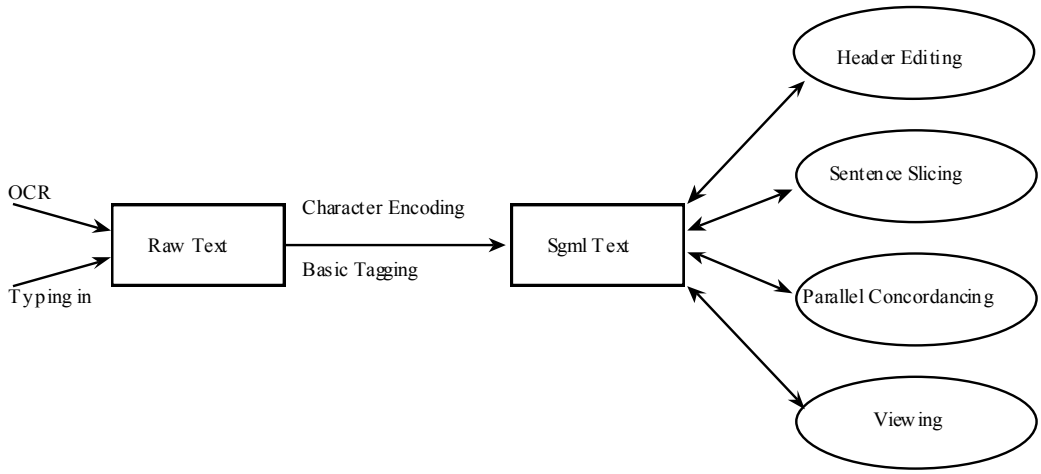

Figure 3.

Towards a user-driven interface

All the preceding corpus management is designed to allow a variety of users access to the texts of the corpus. As has been mentioned, the impetus for the project came from lecturers, and their main specification was for an interactive interface which could be used, at different levels, by student and teacher alike. Neither group must see anything but the plain text, either at entry point or at user point. Accordingly, all coding and decoding is handled by the programs so that there are no distractions in terms of entering on a keyboard, post-editing OCR texts or reading the results on a screen. Likewise, it is intended that the amalgamation of searching and sorting facilities will be handled by the program, allowing the user to enter simply the word or words required.

To allow maximum multilingual flexibility, there is provision for the selection of the working language, the language which will be used for primary searching and the target language. These are not, of course, mutually exclusive. It may be beneficial for the help system to be available in the home language, for instance, but experienced users may prefer to work entirely in the search language.

The degree of selectivity afforded by the TEI header will allow the required selection of subsets to include, for example, only original texts in the designated search or target language, to attempt to ensure that all translations are only one step away from the original.

The interface offers single word, multiple word and context word options to allow synonyms, antonyms, phrasal patterns and the like to be explored. These are standard on monolingual concordancers, but it is here that the sentence id information will be working hardest in collecting the sentence or sentences which go with the occurrence of the search criteria in the the target language. As has been mentioned, the primary screen will probably still be the KWIC concordance line, but from there the user will be able to move into parallel sentences, parallel paragraphs or full screen text in either search or target language. Again, it is the tree system of TEI/SGML which makes the meeting of this objective across a variety of languages an immediate possibility.

The other main element required by the lecturers is the selection of references for the creation of test materials. It is intended that it will be possible for student users to make use of some of these facilities to create on-screen tests for themselves. The tests provided include a gap-filler, creating a gap where the search word(s) appear in the sentence, Cloze, where complete words are removed at regular intervals which can be specified by the user, and a C-test, where the second half of each word is removed. The facility to restrict the selection to certain of the display items is also needed. Exercises constructed in this way can be saved. Because the corpus is closed, the unique sentence identifiers can be saved also to allow instant access back into the full text from any exercise.

One incidental benefit of the sentence id system is that searching does not have to be restricted to the search language. It makes it possible to specify inclusions or exclusions for the target language to restrict the content of the material offered for examination, which may become more important as the corpus grows. It may be possible to distinguish concessive use of 'while' in English by excluding the target language word(s) which express its temporal use, so saving some editing of the search language.

Although we are still at an early stage in the project, it does appear that the adoption of the TEI standard will allow the users to obtain a good deal of what they want whilst leaving the corpus in a controlled state.

Towards flexible tools and resource

Despite the considerable amount of work implied for anyone who wishes to conform to such guidelines as those produced by the Text Encoding Initiative, it appeared to us that this was definitely worth the effort. Indeed, as a norm, it makes it possible for a given electronic text to be transferred from one scholar to another without spending hours in defining an interchange format and writing tools for transcription. Besides, it offers the possibility to incrementally add new information to a given text without any loss of generality. We have seen how natural it was to add alignment information to a text, but in the context of second language teaching, there is always a possibility to add some specific syntactic or rethorical elements on the basis of the different sets of tags defined within the TEI. It is clear however, that such encodings have to be accompanied by a clearly defined set of tools which will effectively give a semantics to the corresponding marks.

Acknowledgements

The work described in this paper is part of Lingua project n° 93-09/1245/F-VB funded by the European Union.

References

Fu, K.S., 1982, *Syntactic Pattern Recognition and applications*, Prentice-Hall, N.J.

Gale, William A. & Church, Kenneth W., "A Program for Aligning Sentences in Bilingual Corpora", Technical Memorandum, AT&T Bell Laboratories, August 15, 1990. *in Proc. of the 29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, 1991, pp.169-176.

Isabelle, Pierre et alii, "Translation Analysis and Translation Automation", *in Proc. of TMI-93*, Kyoto, Japan, July 1993.

Isabelle, Pierre, "Une nouvelle génération d'aides à la traduction et à la terminologie", *Actes du Symposium International Terminologie et Documentation dans la Communication spécialisée*, 1991.

Johns, Tim, 1986, "Micro-concord: a language-learner's research tool", *System* 14/2.

Johns, Tim & King, Philip (eds), Classroom Concordancing, special issue of ELR Journal (New Series) Vol. 4, 1991.

Miclet, Laurent, 1986, *Méthodes structurelles pour la reconnaissance des formes*, Eyrolles, Paris.

Tribble, C. and Jones, G., 1990, *Concordances in the classroom*, Longman.

Appendix: an alignment example.

Source text in file LPPFra.txt

```
<!doctype TEI.2 system "tei2.dtd" [
        <!entity % TEI.prose 'INCLUDE'>
        <!entity % TEI.linking 'INCLUDE'>
        <!entity target.TEI system
"LPPEng.txt">
]>
<TEI.2>
<teiHeader>
<!-- Complete header to be found here -->
</teiHeader>
<text>
<body id=Frenchbody>
<!-- excerpt from the full text - paragraph 10 -->
<p id=p10><s id=p10s1>J&apos;ai donc d&ucirc; choisir un autre m&eacute;tier et j&apos;ai appris &agrave; piloter des avions.</s><s id=p10s2> J&apos;ai vol&eacute; un peu partout dans le monde.</s><s id=p10s3> Et la g&eacute;ographie, c&apos;est exact, m&apos;a beaucoup servi.</s><s id=p10s4> Je savais reconna&icirc;tre, du premier coup d&apos;oeil, la Chine de l&apos;Arizona.</s><s id=p10s5> C&apos;est tr&egrave;s utile, si l&apos;on est &eacute;gar&eacute; pendant la nuit.</s></p>
</body>
<!-- Sentence alignment-->
<!- external pointers to give access to sentences in the target file -->
<xptr id=xptr1 doc=target.TEI from='ID(p10s1)'>
<xptr id=xptr2 doc=target.TEI from='ID(p10s2)'>
<xptr id=xptr3 doc=target.TEI from='ID(p10s3)'>
<xptr id=xptr4 doc=target.TEI from='ID(p10s4)'>
<!-- Linking of two sentences in the French source file -->
<link id=p10s2-3 type=linking targets='p10s2 p10s3'>
<linkGrp type='FR.ENG' domains='Frenchbody Englishbody' targType='s'
targFunc='source target' targOrder=Y evaluate=all>
<!-- Linking of aligned chunks of texts -->
<link targets='p10s1 xptr1'>
<link targets='p10s2-3 xptr2'>
<link targets='p10s4 xptr3'>
<link targets='p10s5 xptr4'>
</linkGrp>
</text>
</TEI.2>
```

Target text in file LPPEng.txt

```
<!doctype TEI.2 system 'tei2.dtd'[
<!entity % TEI.prose 'INCLUDE'>
<!entity % TEI.linking 'INCLUDE'>
]>
<TEI.2>
<teiHeader>
<!-- Complete header to be found here -->
</teiHeader>
<text>
<body id=Englishbody>
<!-- excerpt from the full text - paragraph 10 -->
<p id=p10><s id=p10s1>So then I chose another profession, and learned to pilot aeroplanes.</s><s id=p10s2> I have flown a little over all parts of the world; and it is true that geography has been very useful to me.</s><s id=p10s3> At a glance I can distinguish China from Arizona.</s><s id=p10s4> If one gets lost in the night, such knowledge is valuable.</s></p>
</body>
</text>
</TEI.2>
```